Verwaltung von MS-Stromnetzen

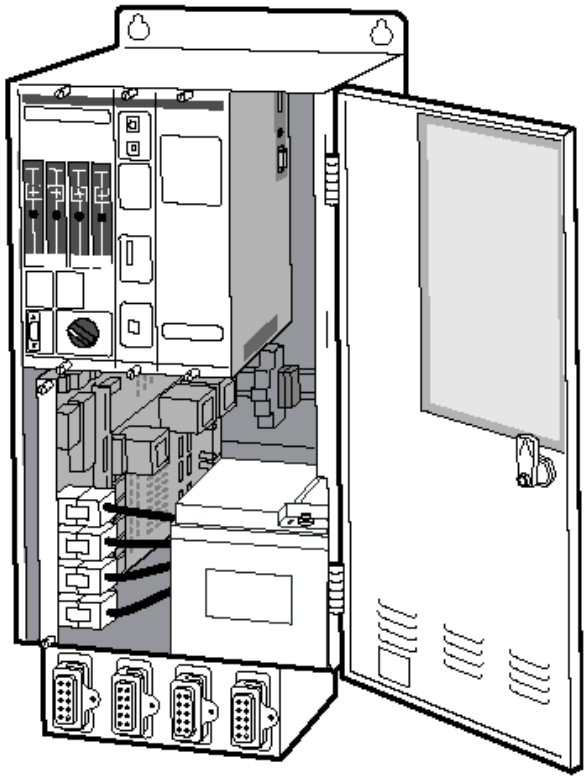
Baureihe Easergy

Flair 200C und T200

Fernüberwachungs- und Steuereinheit für MS-Schaltanlagen

RTU Control und Formeln

Anhang zum Benutzerhandbuch



1 VIRTUELLE SPS 3

1.1 BESCHREIBUNG 3
1.2 INSTALLATION VON RTU CONTROL 5
1.3 EIN PROGRAMM ÜBER RTU CONTROL LADEN 6
1.3.1 Ein Projekt öffnen 6
1.3.2 Anschließen an die SPS und Laden eines Programms 7
1.4 GEMEINSAM GENUTZTE VARIABLEN 8
1.4.1 Gemeinsame Nutzung einer Variablen 8
1.4.2 Verwendung von Arrays 11
1.4.3 Verschiedenes 12
1.5 SPS-EINSTELLUNG 13
1.5.1 SPS-Konfiguration 13
1.5.2 Ein Programm über den integrierten Webserver laden 14
1.5.3 Betriebszustand der SPS 15
1.5.4 Fehlermeldungen 15
1.5.5 Seite „PLC variables“ 16
1.5.6 Programmparameter 16
1.5.7 Peer-to-Peer-Konfigurationsseite 17
1.5.8 SPS-Debug-Seite 19
1.6 BEISPIELE 20

2 BERECHNUNGSFORMELN 21

2.1 EINLEITUNG 21
2.2 SYNTAX UND REGELN 22
2.2.1 Operanden 22
2.2.2 Operatoren 23
2.3 FUNKTIONEN 26
2.3.2 Statistische Funktionen 27
2.3.3 Logische Funktionen 28
2.3.4 Zeit/Datum und Zeitfunktionen: 30
2.3.5 Sonstige Funktionen: 31
2.4 EINSTELLUNGEN 32
2.4.1 Variable erstellen 32
2.4.2 Formeleingabe 33

1 Virtuelle SPS

1.1 Beschreibung

Optional verfügt das T200 und das Flair 200C über eine integrierte virtuelle Maschine (SPS) programmierbare Automatikfunktionen nach IEC 61131-3.

Folgende Anwendungen lassen sich üblicherweise mit einem SPS-Programm erstellen:

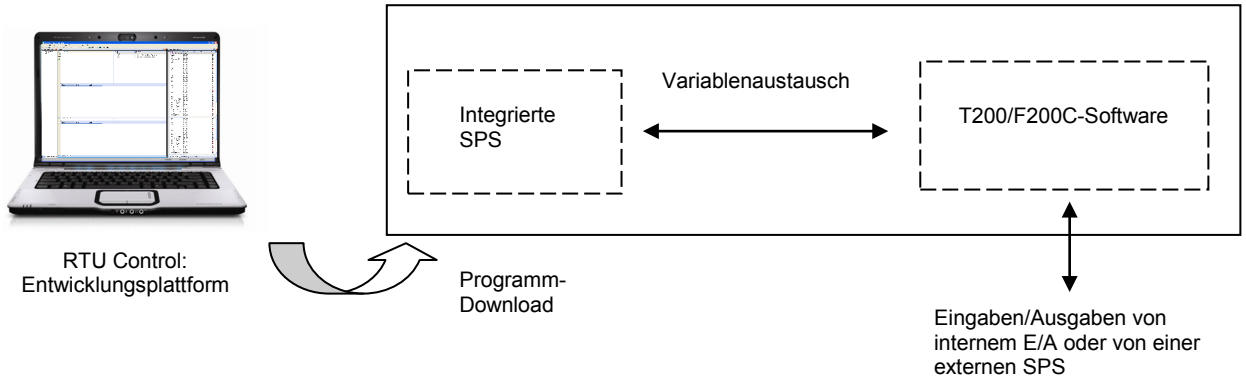
- Automatische Rekonfiguration eines MS-Rings (Self Healing Grid)
- Ähnlicher Automatismus wie der in T200 integrierte Automatismus (ATS, BTA): ATS zwischen 2 T200s oder z. B. bei der Beteiligung von mehr als zwei Kanälen
- Automatischer Lastabwurf
- Beliebige kundenspezifische Automatisierungsfunktionen

Warnung: Im Produkt können nicht gleichzeitig die Option IEC 61131-3 PLC und der vordefinierte Automatismus (ATS, Sectionalizer, ...) aktiviert sein.

Über eine Entwicklungsplattform ist die Erstellung, Simulation und Prüfung von Programmen mithilfe einer der folgenden Sprachen möglich:

- SFC – Sequential Function Chart (AS – Ablaufsprache)
- FBD – Function Block Diagram (FBS – Funktionsbausteinsprache)
- LD – Ladder Diagram (KOP – Kontaktplan)
- ST – Structured Text (ST – Strukturierter Text)
- Il – Instruction List (AWL – Anweisungsliste)

Bei dieser Entwicklungsplattform handelt es sich um die Software RTU Control, die im Lieferumfang des Produkts enthalten ist.
Diese Plattform kann ein Programm erstellen, das von einer virtuellen SPS ausgeführt wird. Die in das T200I integrierte virtuelle SPS tauscht mit der T200I-Software Daten im Lese-/Schreibmodus aus, wobei auf interne Variable zugegriffen werden kann.

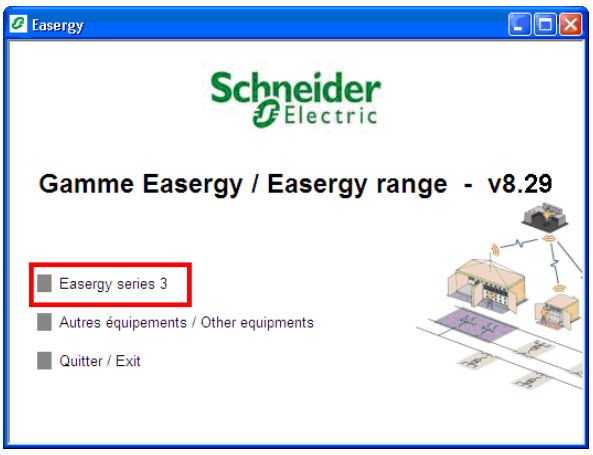


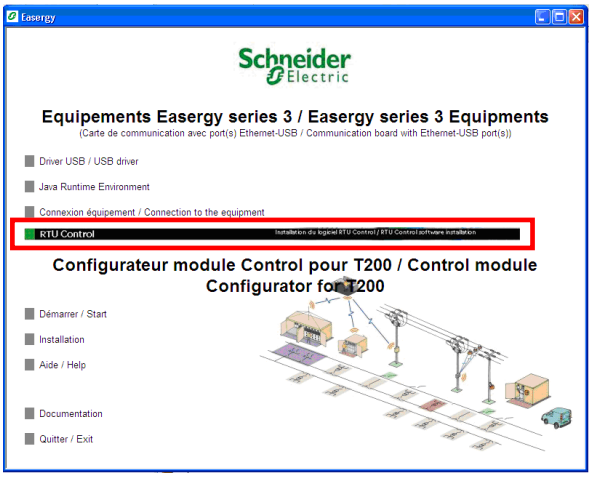
Virtuelle SPS

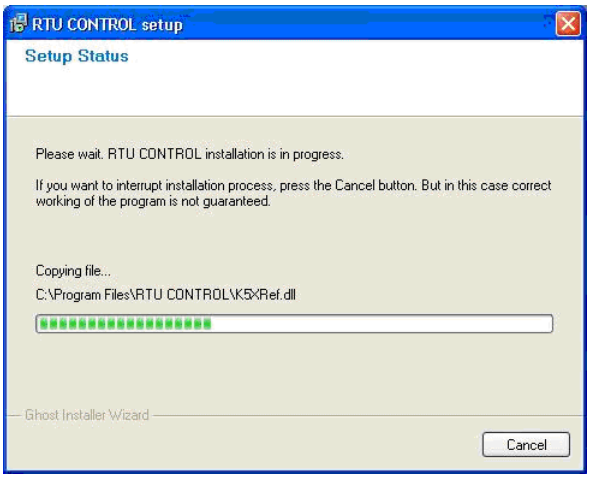
Dieses Dokument enthält keine detaillierten Anweisungen zur Programmierung mit IEC61131-3-Sprachen und zur Verwendung der Plattform. Über RTU Control können Sie auf die entsprechenden Dokumente zugreifen.

In diesem Dokument wird erläutert, wie Sie ein über den Webserver T200/F200C verfügbares SPS-Programm laden und verwenden können. Außerdem werden einführende Anweisungen zur Plattform und zur Verwaltung der Parameter und zugehörigen Variablen erteilt.

1.2 Installation von RTU Control




Easergy
Schneider Electric
Equipements Easergy series 3 / Easergy series 3 Equipments
(Carte de communication avec port(s) Ethernet-USB / Communication board with Ethernet-USB port(s))
Driver USB / USB driver
Java Runtime Environment
Connexion équipement / Connection to the equipment
RTU Control
Configurateur module Control pour T200 / Control module Configurator for T200
Démarrer / Start
Installation
Aide / Help
Documentation
Quitter / Exit


RTU CONTROL setup
Setup Status
Please wait. RTU CONTROL installation is in progress.
If you want to interrupt installation process, press the Cancel button. But in this case correct working of the program is not guaranteed.
Copying file...
C:\Program Files\RTU CONTROL\K5XRef.dll
Ghost Installer Wizard
Cancel

RTU Control ist im Lieferumfang des Geräts enthalten. Die Software befindet sich auf der Easergy CD-ROM.

So installieren Sie die Software von der CD:

- Legen Sie die mitgelieferte CD-ROM ein.
- Die Software auf der CD wird automatisch gestartet. Auf dem Bildschirm wird das Fenster „*Easer range - V8.xx*“ angezeigt.
- Klicken Sie auf den Link „*Easergy series 3*“.
- Klicken Sie in dem nächsten Fenster, das daraufhin angezeigt wird, auf den Link „*RTU CONTROL*“.
- Die Installation der Software startet.
- Führen Sie dann die verschiedenen Installationsschritte aus, ohne die vorgeschlagenen Optionen zu ändern.
- Warten Sie auf das Ende der Installation, und beenden Sie dann das Programm.

Nachdem Sie RTU Control installiert haben, laden Sie ein Programm aus der SPS.

Es gibt zwei Methoden, um ein SPS-Programm in das T200/F200C zu laden:

- **Über die RTU Control Plattform,** vorausgesetzt dass RTU Control auf dem an das T200/F200C angeschlossenen PC installiert ist und dass das Gerät mit gestarteter SPS-Funktion hochgefahren wird.

- **Über den auf dem T200/F200C integrierten Webserver**, vorausgesetzt, dass das SPS-Programm in Form einer Datei mit der Erweiterung *.cod verfügbar ist.

1.3 Ein Programm über RTU Control laden

In diesem Kapitel wird die erste Methode für das Laden eines Programms von der SPS-Software RTU Control beschrieben. Voraussetzung ist, dass RTU Control auf dem an den Webserver des T200/F200C angeschlossenen PC installiert und das entsprechende Projekt verfügbar ist.

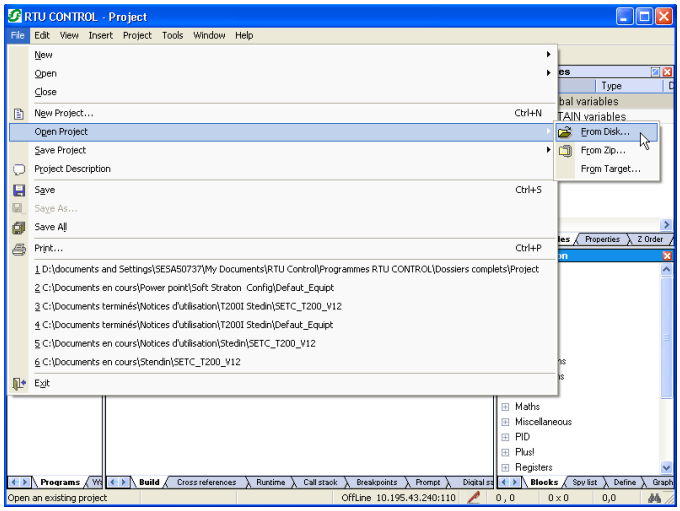
1.3.1 Ein Projekt öffnen

Das Projekt wird von der Programmoberfläche RTU Control aus geöffnet.

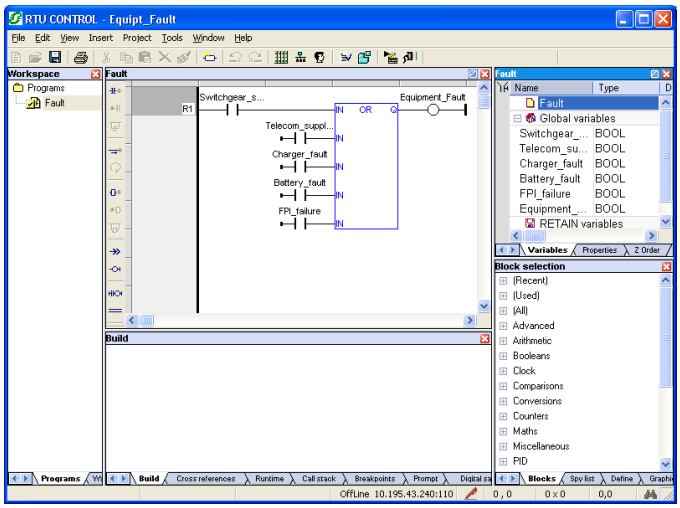
Menu File → Open project → From Disk

Wählen Sie das Projekt aus, das Sie laden möchten.

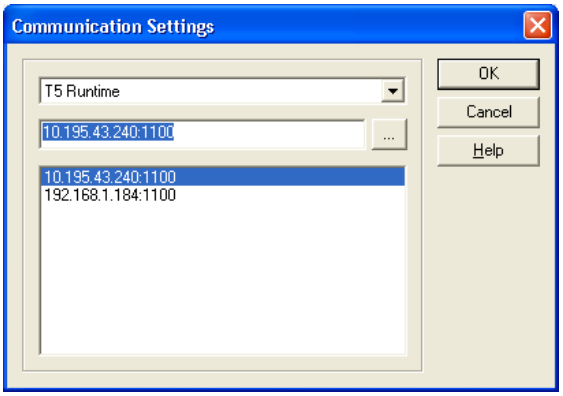
Das Projekt wird in der Workbench geöffnet.

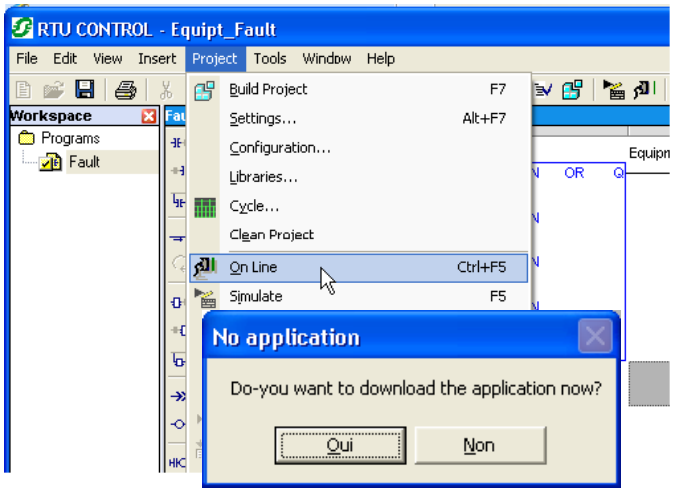


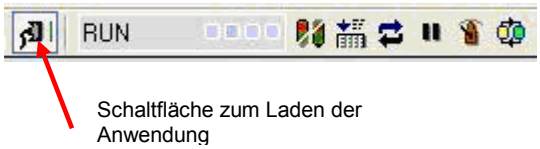
Ein Programm zum Öffnen in RTU Control auswählen



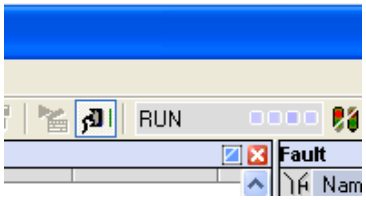
In der Workbench geöffnetes Programm



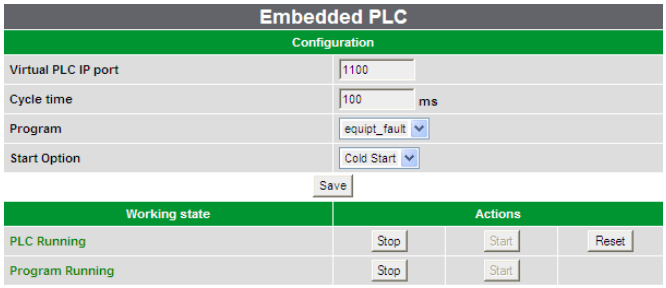




Schaltfläche zum Laden der Anwendung



„Ausführen“-Status in der RTU Control Workbench



„SPS wird ausgeführt“-Status auf der Seite „Embedded PLC“ des Webservers.

1.3.2 Anschließen an die SPS und Laden eines Programms

Sobald das Programm geöffnet ist, konfigurieren Sie die Verbindung zur virtuellen SPS.

Öffnen Sie von der RTU Control Workbench aus den folgenden Link:

Menu Tools → Communication parameters.

- Wählen Sie „T5 Runtime“.
- Geben Sie die IP-Adresse des Geräts ein, gefolgt von der IP-Portnummer, die von der virtuellen Maschine verwendet wird (bereits auf dem Gerät konfiguriert). Die beiden Felder müssen durch das Zeichen „:“ getrennt sein (wenn kein IP-Port angegeben ist, wird 1100 verwendet).
- Das Beispiel zeigt eine Konfiguration für ein T200 mit der IP-Adresse 10.199.45.103 und der IP-Portnummer 1100.
- Klicken Sie auf „OK“.

Um die Workbench mit dem T200/F200C zu verbinden und die Anwendung zum Zielort herunterzuladen klicken Sie auf:

Menu Project / On line / Download application.

Klicken Sie zum Stoppen der Anwendung auf dieselbe Schaltfläche.

Hinweis: Dieser Befehl kann direkt über die Taskleiste ausgeführt werden.

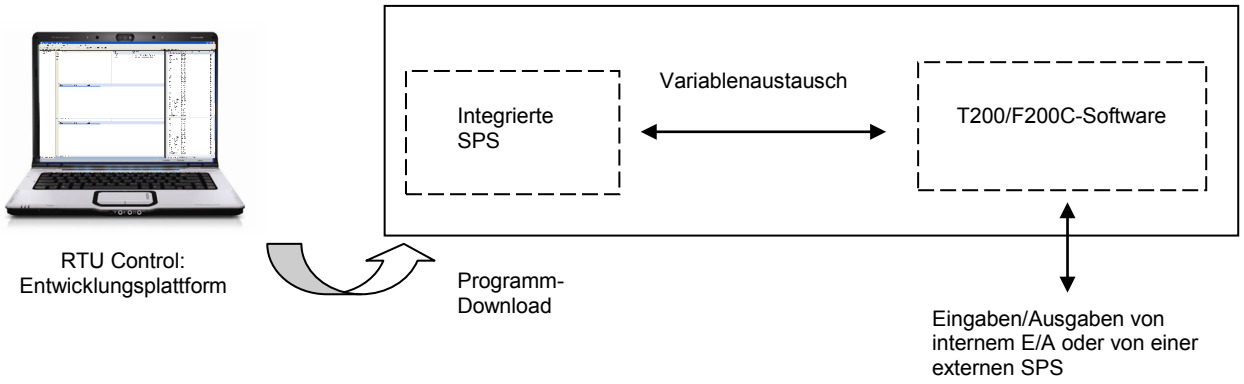
Das Programm kann nun im Gerät verwendet werden. Es wird jetzt mit dem Status „Running“ angezeigt:

- In **RTU Control**
- Auf der Seite **Embedded PLC** des T200/F200C-Webservers

Hinweis: Im Folgenden wird die Seite „Embedded PLC“ und der Zugriff darauf näher erläutert.

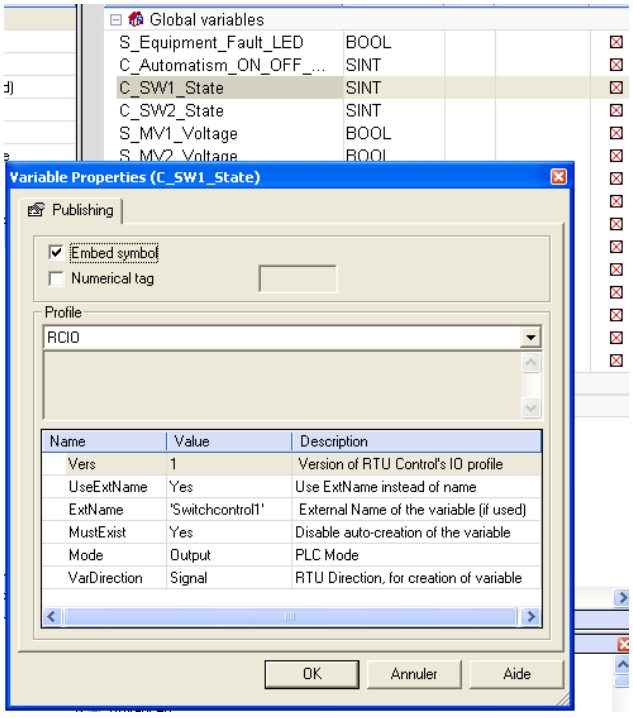
1.4 Gemeinsam genutzte Variablen

Die integrierte SPS nutzt interne Variablen und kann auch gemeinsam mit der T200/F200C-Software auf Variablen zugreifen.
Eine auf der Plattform gemeinsam genutzte Variable wird beim Hochfahren der Anwendung automatisch mit einer T200/F200C-Variablen verknüpft.



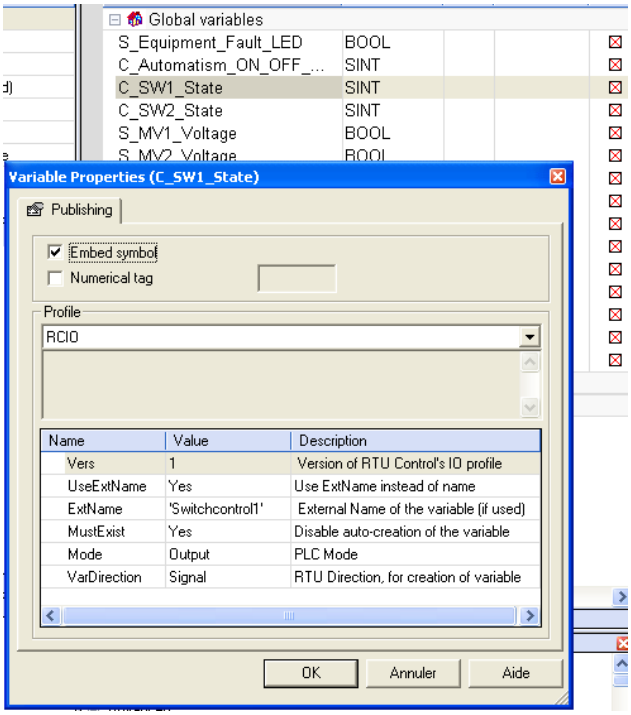
Die gemeinsam genutzten Variablen sind **globale** Variablen. Diese Variablen können überall im Programm verwendet werden.

Der Link zwischen SPS-Variablen und T200/F200C-Variablen kann auf verschiedene Weise hergestellt werden:
- völlig identische Namen oder Verwendung verschiedener Namen
- automatische Erstellung von Variablen in dem T200/F200C oder Verwendung vorhandener Variablen
- bei einer Erstellung Auswahl des Austauschmodus (aus T200/F200C-Perspektive)



1.4.1 Gemeinsame Nutzung einer Variablen

Klicken Sie in der Workbench (vom Gerät getrennt oder mit gestoppten Programm) mit der rechten Maustaste auf eine Variable, und wählen Sie „*properties*“.
So können Sie die ausgewählte Variable zusammen mit der T200/F200C-Software verwenden:
- Aktivieren Sie das Kästchen „*Embed symbol*“.
- Wählen Sie aus der Profilliste das Profil „RCIO“ aus
- Geben Sie je nach gewünschtem Linkmodus die folgenden Parameter an:



UseExtName: Hier haben Sie die Wahl zwischen dem in der Workbench angegebenen Namen der Variablen, wenn er von dem Namen in T200/F200C abweicht („Yes“), oder einer erzwungenen Verwendung desselben Namens wie der Name der in T200/F200C definierten Variablen (Standardwert ist „No“).

ExtName: Name der T200/F200C-Variablen, die der Workbench-Variablen zugeordnet ist. Mit diesem Parameter können Sie die Verbindung zwischen der Variablen der T200/F200C und der entsprechenden Variablen in der Workbench herstellen, wenn sie verschiedene Namen haben (wird nur verwendet, wenn „UseExtName“ = „Yes“ ist). Die Feldlänge ist auf 15 Zeichen begrenzt.

MustExist: Auswahl zur Autorisierung für die Erstellung einer Variablen (Standardwert „No“) oder erzwungenen Verwendung bereits vorhandener Variablen.
Beim Start des Programms prüft das T200/F200C alle gemeinsam genutzten Variablen. Wenn die zugehörige Variable vorhanden ist, wird der Link erstellt. Wenn die Variable nicht vorhanden ist, kann ihre Erstellung durch das T200/F200C autorisiert oder verboten werden.
Dieser Parameter ermöglicht eine bessere Steuerung, wenn eine Verknüpfung mit einer vorhandenen Variablen vorgenommen werden soll.

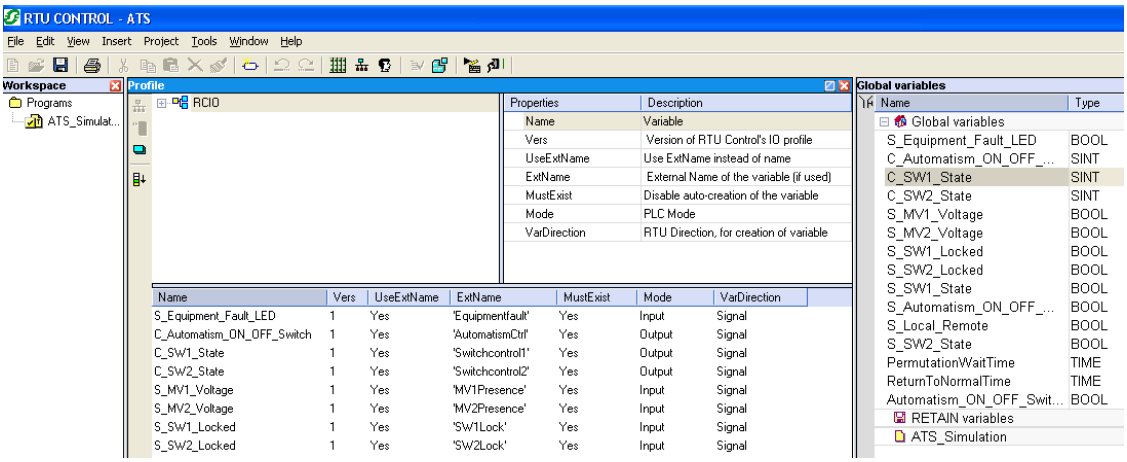
Modus: [Eingabe/Ausgabe] Datenaustauschmodus.
Der Modus wird verwendet, um festzustellen, ob Variablen aus der Perspektive der SPS Lese- oder Lese/Schreib-Variablen sind. In der SPS-Anwendung als Eingaben definierte Variablen sind Lese-Variablen, d. h. sie können nur von der RTU-Firmware geändert werden. Als Ausgaben definierte Variablen sind Lese/Schreib-Variablen. Sowohl für Ein- als auch Ausgaben wird der aktuelle Wert am Anfang jedes SPS-Zyklus gelesen. Nur Ausgangsvariablen werden am Ende des SPS-Zyklus aktualisiert.
Bitte beachten Sie, dass bestimmte Variablen (wie TSS) automatisch Lese-Variablen sind.

VarDirection: [Signal/Befehl] Variablentyp aus T200/F200C-Perspektive. Wird verwendet, wenn eine Variable von dem T200/F200C erstellt wird (MustExist = No und entsprechende Variable ist nicht vorhanden).
Als „Signal“ definierte Variable wird als Eingabe (DI oder AI) erstellt. Als „Befehl“ definierte Variable wird als Ausgabe (DO oder AO) erstellt.
Die Befehle können von dem Benutzer über die Webschnittstelle oder über das Protokoll gesteuert werten (Beispiel: Parameter, Sollwert, Option).

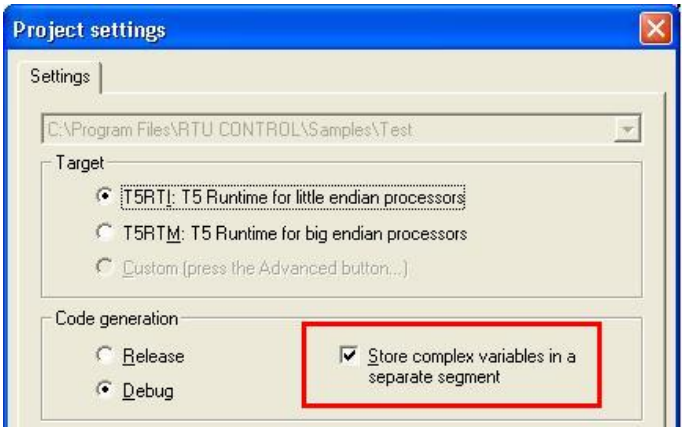
Warnung: Das Konzept der Eingabe/Ausgabe aus T200/F200C-Perspektive unterscheidet sich von dem Konzept aus SPS-Perspektive, wo ein Parameter beispielsweise die „Eingabe“ eines Funktionsblocks ist.

Nach Änderungen an der Variablenkonfiguration muss das Programm neu aufgebaut und wieder in das Gerät geladen werden, damit die Änderung wirksam wird.

Die Einstellungen der gemeinsam genutzten Variablen werden über die Browserprofile verwaltet (Symbol „Profile öffnen“).



Über diesen Explorer können Sie eine globale Ansicht der mit dem RCIO-Profil verknüpften Profile aufrufen und sie ändern.
Mithilfe der „Drag and Drop“-Funktion lässt sich eine Variable einfach mit einem Profil verknüpfen.

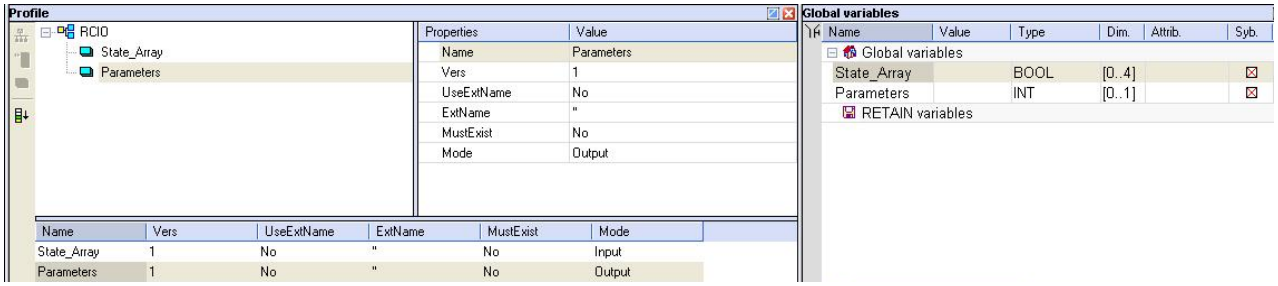


1.4.2 Verwendung von Arrays

Die SPS kann intern komplexe Arrays mit Variablen oder Strukturen verwenden.
Bei gemeinsam genutzten Variablen sind ausschließlich eindimensionale Arrays für einfache Datentypen zulässig. In diesem Fall muss die Option „*Store complex variables in a separate segment*“ in den Optionen des Projekts aktiviert werden (Project\Settings…).

Ein Array aus N-Variablen wird mit N-Variablen in der T200/F200C-Software verknüpft. Der Name dieser Variablen ist NameArray_@x mit x von 1 bis N.

Beispiel: Erstellung von Arrays „State_Array“ aus 5 booleschen Werten (BOOL) und „Parameter“ aus 2 Ganzzahlen (INT)



In der T200/F200C-Software erstellte Variablen sind State_Array_@1 bis State_Array_@5 und Parameters_@1 bis Parameter_@2.

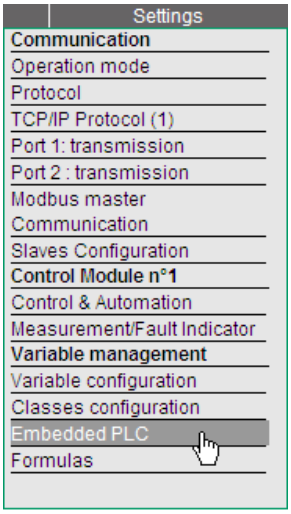
	STATE_ARRAY_@1		DI18	-
	STATE_ARRAY_@2		DI19	-
	STATE_ARRAY_@3		DI20	-
	STATE_ARRAY_@4		DI21	-
	STATE_ARRAY_@5		DI22	-
	PARAMETERS_@1		AO13	-

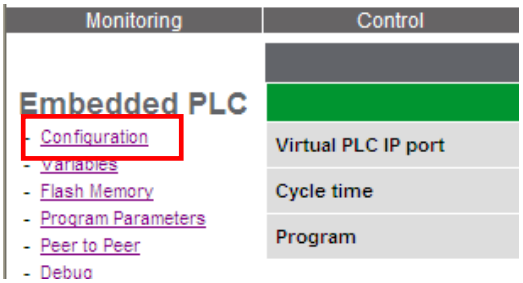
1.4.3 Verschiedenes

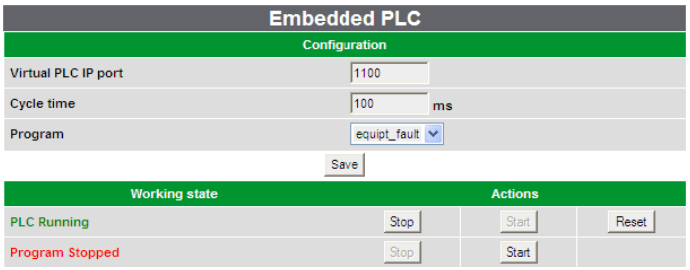
Hinweise zu den Namen der Variablen:
- Verwenden Sie nur Großbuchstaben.
- Geben Sie zwei verschiedenen Variablen nicht denselben Namen.
- Verwenden Sie keine Ganzzahl als Name (zum Beispiel „5“)

Variablentypen:
Das T200/F200C unterstützt die folgenden Typen: BOOL, SINT, INT, DINT, BYTE, WORD, REAL.
Warnung: Typ muss kompatibel sein.
BOOL <--> TSS, TSL, DI oder DO
SINT, INT, DINT, WORD, DWORD, REAL <--> TM, AI oder AO
DINT <--> Zähler
Wichtig: Die TSD/TCD sind keine binären, sondern numerische Variablen (BYTE) mit drei möglichen Zuständen:

TSD/TCD-Wert	Zustand
0	Undefiniert
1	Aus
2	Ein







Konfigurationsseite für integrierte SPS

1.5 SPS-Einstellung

Im gleichen Abschnitt „SPS“ des Webservers sind auch andere Seiten mit der SPS verknüpft, vor allem die Seiten in Bezug auf die Konfigurationsparameter.

1.5.1 SPS-Konfiguration

Klicken Sie auf den Link „*Parameter*“, um die Konfigurationsseite für die SPS-Parameter aufzurufen.

Diese Seite enthält Informationen über den Betriebszustand der virtuellen Maschine und des zugehörigen Programms sowie mögliche Verfügbarkeits- oder Fehlermeldungen.

1.5.1.1 IP-Port der virtuellen SPS:

Nummer des IP-Ports, der von dem SPS-Programm verwendet wird (RTU Control).
Standardwert: 1100

1.5.1.2 Zykluszeit:

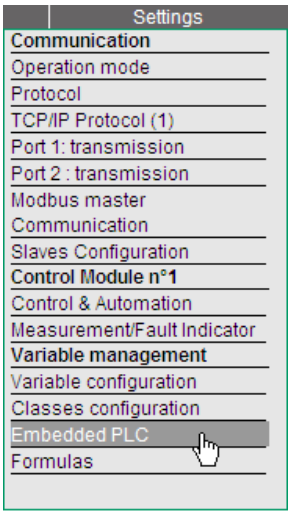
Dieser Wert gibt in Millisekunden die Dauer des Zyklus der virtuellen Maschine an. Dieser Wert in ms muss zwischen 100 und 10000 ms (10 Sekunden) liegen.

1.5.1.3 Programm:

Name des geladenen und von der virtuellen Maschine ausgeführten Programms, einschließlich aller Programme im Flash-Speicher.
Das T200/F200C unterstützt mehrere Programme im Speicher, es kann jedoch immer nur eines ausgeführt werden.

(Mehr Informationen finden Sie im Kapitel „*Ein Programm über den integrierten Webserver laden*“.)

Hinweis: Die Option zum Starten der SPS ist nicht konfigurierbar. Dieses Hochfahren wird immer als „Kaltstart“ ausgeführt.
Entsprechend werden alle internen Variablen der SPS beim Hochfahren auf ihren ursprünglichen Wert zurückgesetzt.
Die Werte von „Retain”-Variablen werden nicht verwendet.

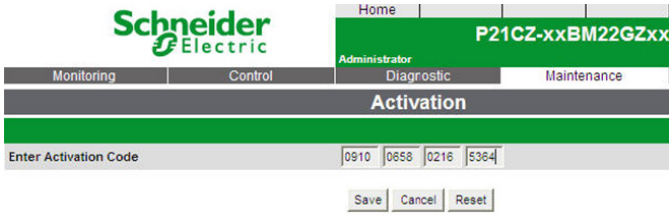


1.5.2 Ein Programm über den integrierten Webserver laden

Dieses Kapitel erläutert die zweite Methode zum Laden eines SPS-Programms von dem im T200/F200C integrierten Webserver.

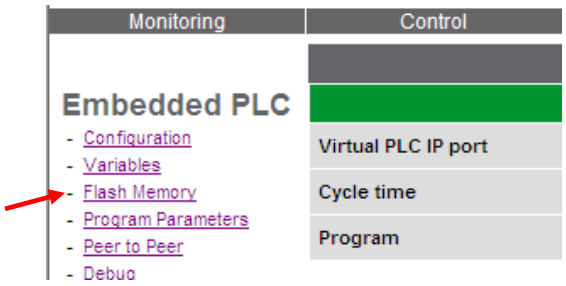
Um das Programm zu laden, benötigen Sie zuerst die *.cod-Datei.

Um auf die Seite „Embedded PLC“ des integrierten Webservers zuzugreifen, klicken Sie auf den entsprechenden Link („*Embedded PLC*“).



Seite für die Aktivierung von RTU Control

Wenn RTU Control noch nicht auf dem Gerät aktiviert wurde, geben Sie den Aktivierungscode (von Schneider Telecontrol) ein.

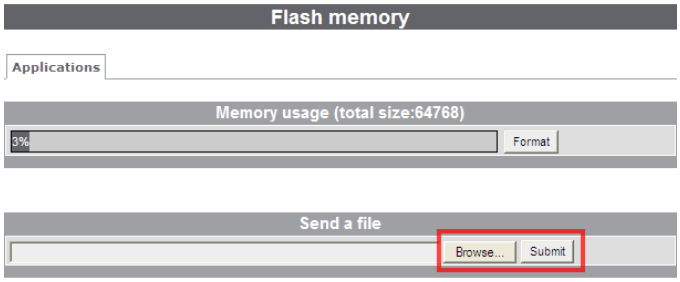


Klicken Sie dann auf den Link „*Flash Memory*“, um auf die SPS-Datei und die Seite für die Speicherverwaltung zuzugreifen.

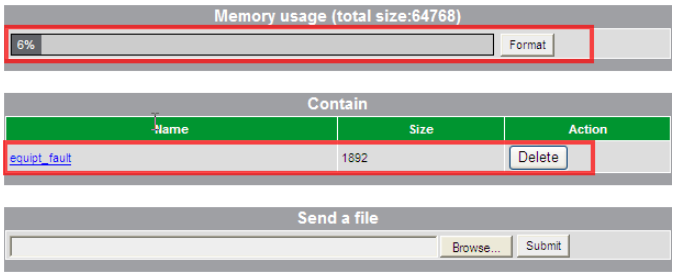
Diese Seite verwaltet die Programme der in den Flash-Speicher des Geräts hochzuladenden oder bereits hochgeladenen Anwendung.

Klicken Sie auf die Schaltfläche „*Browse*“, um den Speicherort für die SPS-Programmdatei festzulegen (Erweiterung „*.cod“), die geladen werden soll.

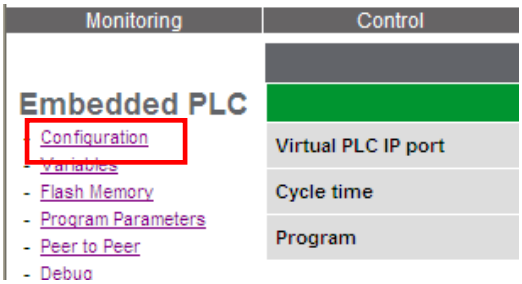
Klicken Sie dann auf die Schaltfläche „*Submit*“, um das Programm in der virtuellen SPS anzuerkennen.



Seite „Flash memory“



Seite „Flash memory“



Working state	Actions		
PLC Stopped	Stop	Start	Reset

SPS gestoppt

Working state	Actions		
PLC Running	Stop	Start	Reset
Program Stopped	Stop	Start	

SPS läuft, Programm gestoppt

Working state	Actions		
PLC Running	Stop	Start	Reset
Program Running	Stop	Start	

SPS und Programm laufen

Errors
Error Loading Program
Error Saving Program
Error Saving code change

Verschiedene Fehlermeldungen

Das Programm kann nun verwendet werden.

Der Name des Programms wird nun in dem Abschnitt „*Contain*“ mit Angabe der Dateigröße angezeigt.

Das Programm kann in der SPS über die Schaltfläche „*Delete*“ gelöscht werden.

Diese Datei kann durch Klicken auf den Namen des Programms auf die Festplatte des PCs mit der Erweiterung *.cod geladen werden.

Die von dem Programm verwendete Speichergröße wird auch als Prozentzahl angezeigt.

Es wird empfohlen, den Speicher vor der ersten Verwendung mithilfe der Schaltfläche „*Format*“ zu formatieren.

1.5.3 Betriebszustand der SPS

1.5.3.1 SPS-Status

Auf der Seite „*Embedded PLC/Configuration*“ gibt die erste Zeile des Abschnitts „*Working state*“ an, ob das Modul läuft oder gestoppt ist.
Das Modul kann dann mithilfe der entsprechenden Schaltflächen gestartet oder gestoppt werden.
Dieser Parameter wird gespeichert (bei einem RTU-Neustart ist das Modul in seinem letzten Zustand).

Hinweis: Damit die Verbindung mit der virtuellen Maschine möglich ist, muss das Modul laufen.

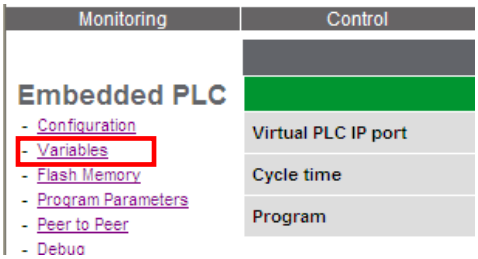
1.5.3.2 Zustand der virtuellen Maschine

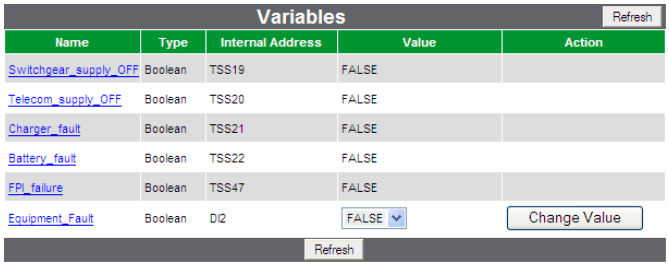
Wenn das Modul läuft, zeigt eine zweite Zeile den Zustand der virtuellen Maschine an, die das Programm ausführt.
Das Programm kann dann mithilfe der Schaltflächen auf der Benutzeroberfläche gestartet oder gestoppt werden.

Hinweis: Je nach Anzahl der Variablen im Programm kann die Verzögerung beim Hochfahren länger oder kürzer sein.

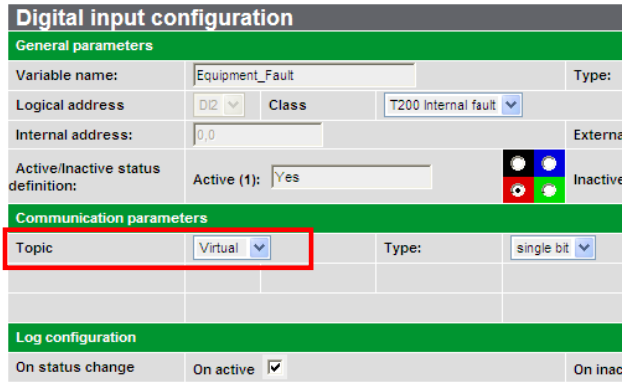
1.5.4 Fehlermeldungen

Wenn der Start des Moduls oder des Programms fehlgeschlagen ist, kann/können eine oder mehrere Fehlernachrichten angezeigt werden.

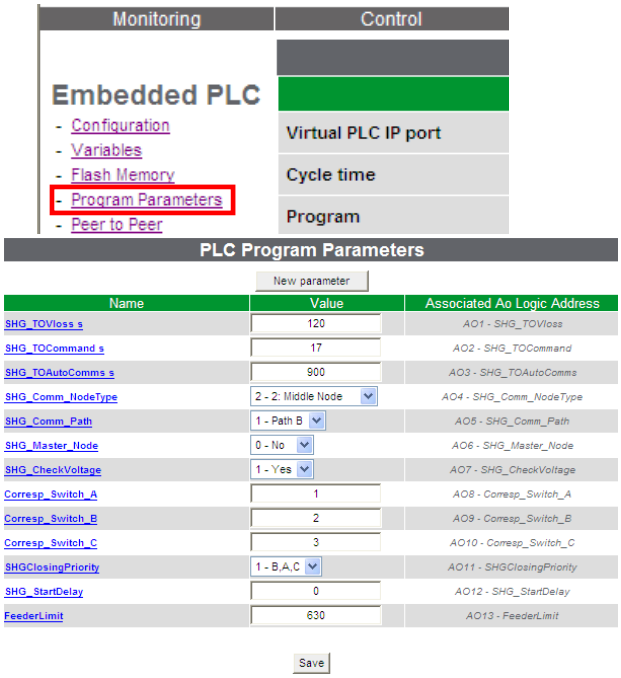




Seite „Variablen“



Konfigurationsseite für eine virtuelle Variable



Beispiel für SPS-Parameter

1.5.5 Seite „PLC variables“

Über die Seite *„Variables“* des Menüs „*Embedded PLC*“ können die internen Variablen verwaltet werden, die für den internen Austausch zwischen SPS und dem T200/F200C-Gerät verwendet werden.

Folgende Variablen können gemeinsam genutzt werden:
- **Vorhandene** Variablen des T200/F200C, die für eine gemeinsame Nutzung mit RTU Control definiert wurden, weil sie im SPS-Programm verwendet werden.
- **Virtuelle** Variablen, die speziell für RTU Control erstellt wurden.

Der Wert einer virtuellen Variablen kann bei Bedarf auf dieser Seite über die Schaltfläche „*Change Value*“ geändert werden.

Ein Klick auf den Namen der Variablen ermöglicht den Zugriff auf die Konfigurationsseite für diese Variable.
Virtuelle Variablen werden von den anderen Variablen durch einen Topic-Wert= „*Virtual*“ unterschieden.

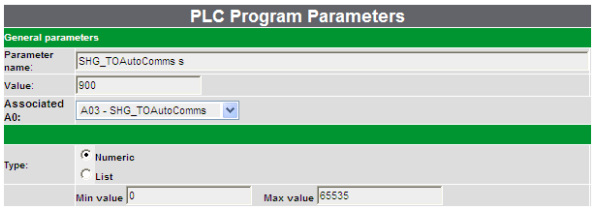
Bis auf den Topic-Wert ist die Konfiguration dieser Variablen identisch wie die der anderen Variablentypen des T200/F200C. Siehe Abschnitt 2.4.1 „Variablen erstellen“.

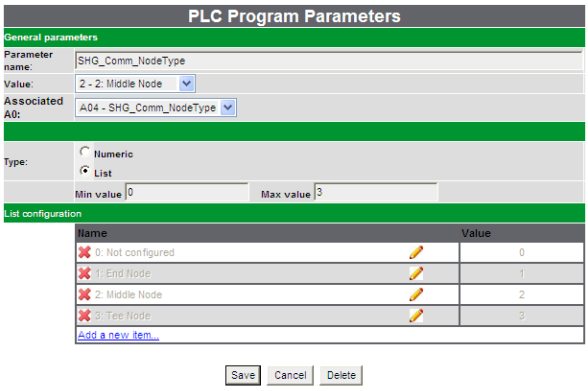
1.5.6 Programmparameter

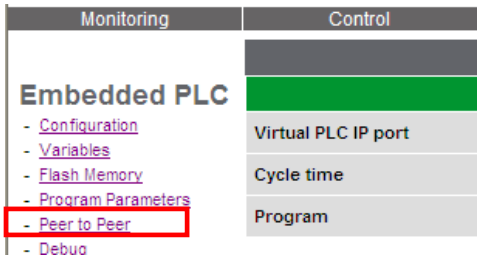
Mit dieser Seite kann ein Parametersatz, der für die SPS-Anwendung gültig ist, erstellt und geändert werden.
Die Parameterwerte werden zusammen mit der Easergy-Konfiguration gespeichert, ohne dass die SPS-Anwendung geändert werden muss. Diese Parameter können innerhalb eines festgelegten Bereichs numerisch sein oder Teil einer Liste, die einen Text einem numerischen Wert zuordnet.

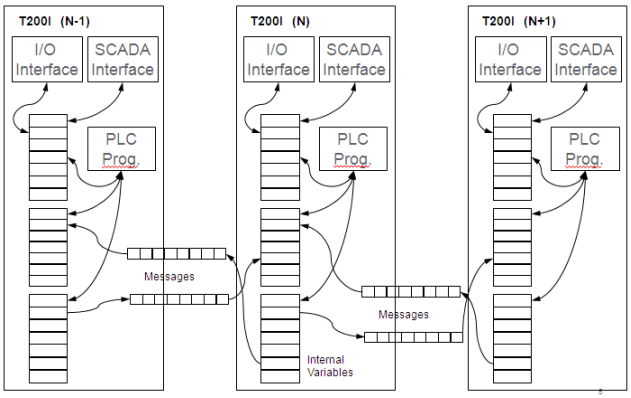
Um die Parameterwerte auf die SPS-Anwendung zu übertragen, müssen virtuelle AO-Variablen definiert werden. Jeder Parameter ist mit einem AO verknüpft, der als gemeinsam genutzte I/O-Variable im SPS-Programm definiert werden muss.

Der SPS-Parameterwert ist den AOs zugeschrieben:
- wenn die RTU hochfährt (aus technischer Sicht beim Hochfahren des Threads der virtuellen SPS-Maschine)
- nachdem neue Werte in die Parameter-Webseite der SPS eingegeben wurden
- nachdem das System eine Änderung in der Konfiguration des MODBUS-Masters und/oder der virtuellen Variablen erkennt









Es können bis zu 20 Parameter definiert werden. Für jeden können die folgenden Parameter definiert werden:

- Name
- Wert
- Zugehöriger AO (der AO muss vorher definiert werden)
- Typ „Numerisch" oder „Liste"
- möglicher Wertebereich (Min/Max)
- für einen Parameter des Typs „List" fügen Sie die möglichen Punkte der Liste hinzu und bearbeiten sie.

1.5.7 Peer-to-Peer-Konfigurationsseite

Die Peer-to-Peer-Funktion ermöglicht eine direkte „Seitwärts"-Kommunikation zwischen zwei oder mehreren RTUs, unabhängig von der „Aufwärts"-Kommunikation mit einem SCADA-System.
Die RTUs müssten im gleichen TCP/IP-Netzwerk installiert sein (Ethernet oder GPRS/UMTS).

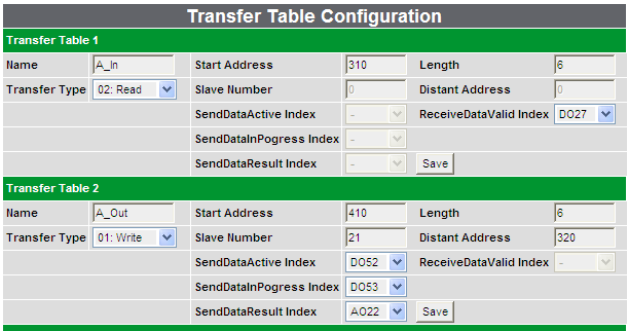
Diese Seite zeigt die Konfiguration der Übertragungstabellen, die zum Versenden von Nachrichten zwischen zwei benachbarten Geräten verwendet werden.

Peer-to-Peer-Prinzip:
Die Peer-to-Peer-Kommunikation kann die Werte einer Reihe von Variablen von einer RTU zur anderen mithilfe des MODBUS TCP-Protokolls und des Befehls „write multiple registers!" kopieren. Dies gleicht dem Konzept eines „Datensatzes" in anderen Protokollen.

Jeder Datensatz wird in beiden Peer-Geräten als Austauschtabelle definiert. Beim Sendegerät wird die Tabelle als „Schreib"-Typ definiert, während sie beim Empfangsgerät als „Lese"-Typ definiert wird. Jede Austauschtabelle definiert einen Bereich von MODBUS-Adressen. Eine Variable gehört zu dem Datensatz, wenn ihre „Peer-to-Peer"-Adresse innerhalb des Bereichs liegt.

Bei „Schreib"-Tabellen sind andere Variablen möglich. Bei „Lese"-Tabellen müssen die Variablen entweder vom Typ DO oder AO sein.

Das SPS-Programm kann das Versenden der „Lese"-Tabellen starten und erhält eine Warnung, wenn die „Lese"-Tabellen aktualisiert werden.
Der Prozess kann weitere Variablen verwenden, um den Erfolg oder das Fehlschlagen einer Nachricht anzuzeigen. Diese können von dem SPS-Programm verwendet werden, um bei Bedarf Nachrichten erneut zu senden.



Konfiguration:
Es können bis zu 6 Tabellen definiert werden.
Für jede Tabelle:

- Name
- Übertragungstyp: „Inaktiv", „Schreiben" oder „Lesen"
- Startadresse: lokale MODBUS-Adresse entsprechend der ersten Variablen der Tabelle
- Länge: Anzahl der 16-Bit-MODBUS-Register Maximal 6 Digitale Variablen werden auf einem einzigen Bit abgebildet. Analoge Variablen verwenden ein 16-Bit-Register. Der „Längen"-Wert muss in der „Schreib"-Tabelle und in der „Lese"-Tabelle gleich sein.

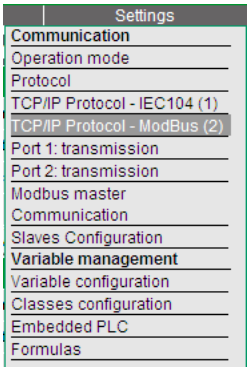
Für den „Schreib"-Typ:

- Slave-Nummer: entspricht dem Slave-Index auf der Slave-Konfigurationsseite (wo seine IP-Adresse definiert wird)
- Externe Adresse: MODBUS-Adresse auf dem Slave, wo die Tabelle geschrieben wird
- SendDataActive-Index: DO-Index (Steuervariable). Dies ist der Index der DO-Variablen, die von der SPS-Anwendung zu aktivieren ist, um die Firmware anzuweisen, den Inhalt der Tabelle in das Ferngerät zu schreiben. Wird von der Firmware automatisch zurückgesetzt, wenn die Übertragung läuft.
- SendDataInProgress DO-Index (Steuervariable). Dies ist der Index der DO-Variablen, der anzeigt, dass der Schreibprozess für diese Tabelle läuft. Wird bei Abschluss der Übertragung oder Zeitüberschreitung automatisch zurückgestellt. Hilfreich für Inbetriebnahme
- SendDataResult-Index: Index einer AO (Steuervariable), die den Status der Übertragung anzeigt.
  Die AO kann folgende Werte haben:
  0: Operation erfolgreich
  1: Ungültige MODBUS-Funktion
  2: Ungültige Adresse
  3: Ungültige Daten
  15: Kommunikationsfehler - Problem bei der Erstellung einer TCP-Verbindung
  16: Kommunikationsfehler - Zeitüberschreitung bei Befehl oder Antwort
  20: Gesetzt, wenn eine Nachricht im Gang ist

  Die Codes 1, 2 und 3 geben an, dass die Nachricht erfolgreich gesendet wurde, dass sie jedoch von dem Empfangsgerät zurückgewiesen wurde. Dies bedeutet normalerweise einen Konfigurationsfehler.

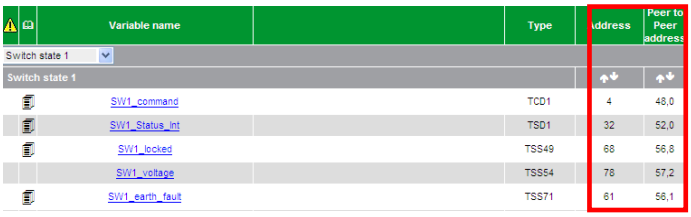
Für den „Lese"-Typ:

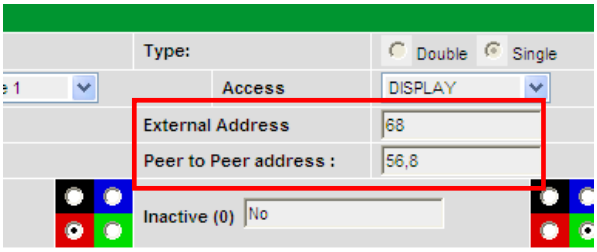
- ReceiveDataValid-Index: DO-Index (Steuervariable). Dies ist der Index einer DO-Variablen, die gesetzt wird, wenn von einem Ferngerät neue Daten geschrieben wurden. Sie sollte von der SPS-Anwendung zurückgesetzt werden.



Protokolle und Variablenadressen
Da die Peer-to-Peer-Funktion das MODBUS TCP-Protokoll verwendet, sind sowohl der MODBUS-Master als auch das MODBUS TCP-Slave-Protokoll neben dem Haupt-Slave-Protokoll (IEC101/104 oder DNP3/DNP3 IP) stets auf dem Gerät aktiviert.

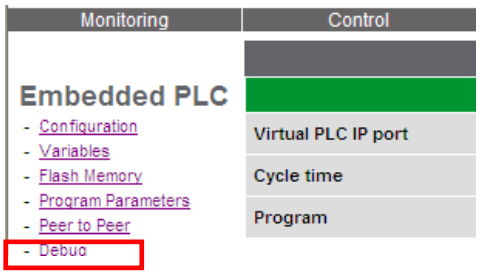
Jedes dieser Protokolle hat seine eigenen Webseiten für die Konfiguration.





Für jede Variable wird von dem Hauptprotokoll die „External address" verwendet, während für die Peer-to-Peer-Funktion die „Peer-to-Peer-Adresse" im MODBUS TCP verwendet wird.

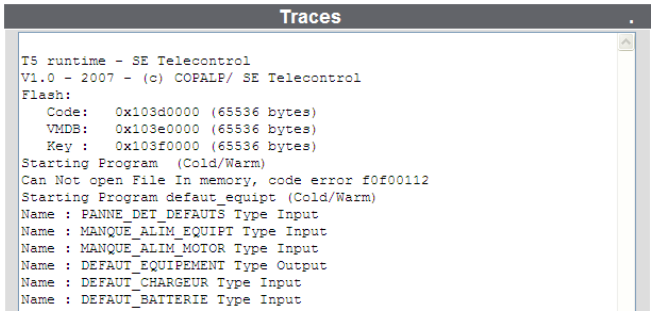
Wenn das Hauptprotokoll MODBUS/MODBUS TCP ist, werden dieselben Adressen (externen Adressen) sowohl für die Scada-Kommunikation als auch die Peer-to-Peer-Kommunikation verwendet.



1.5.8 SPS-Debug-Seite

Die SPS-Trace kann verwendet werden, um zu prüfen, ob RTU Control ordnungsgemäß läuft und die Variablen ohne Fehler in das Gerät geladen wurden.

Beim normalen Betrieb ist eine Überwachung der Programmausführung nicht erforderlich. Nur bei anormalem Verhalten der SPS kann eine Überwachung sinnvoll sein.


Traces
T5 runtime - SE Telecontrol
V1.0 - 2007 - (c) COPALP/ SE Telecontrol
Flash:
Code: 0x103d0000 (65536 bytes)
VMDB: 0x103e0000 (65536 bytes)
Key : 0x103f0000 (65536 bytes)
Starting Program (Cold/Warm)
Can Not open File In memory, code error f0f00112
Starting Program defaut_equipt (Cold/Warm)
Name : PANNE_DET_DEFAUTS Type Input
Name : MANQUE_ALIM_EQUIPT Type Input
Name : MANQUE_ALIM_MOTOR Type Input
Name : DEFAUT_EQUIPEMENT Type Output
Name : DEFAUT_CHARGEUR Type Input
Name : DEFAUT_BATTERIE Type Input

1.6 Beispiele

Zur Veranschaulichung werden einige Projekte als Beispiel angeführt. Sie gehören zur Installation von RTU Control und befinden sich im Ordner „RTU Control\Samples\Easergy”.

Bei diesen Projekten kann beispielsweise der Link zwischen RTU Control und den T200-Variablen sowie die Verwaltung von TSD/TCD angezeigt werden. Sie umfassen die entsprechenden T200- oder F200C-Konfigurationen in einem speziellen Ordner.

Diese Projekte dienen nur zur Veranschaulichung und werden nicht vom technischen Kundendienst unterstützt.

Schneider Electric übernimmt keine Verantwortung für jegliche Folgen, die sich aus der Verwendung dieser Programme ergeben.

2 Berechnungsformeln

2.1 Einleitung

Im Gegensatz zu der SPS, die eine Option für das T200 und das Flair 200C ist, werden die Berechnungsformeln standardmäßig mit der Grundausführung des T200 geliefert.

Die Berechnungsformeln ermöglichen arithmetische und kombinatorische Verknüpfungen zwischen Variablen.

Die Formeln dienen folgendem Zweck:

- Ausführung mathematischer Berechnungen auf der Grundlage von physischen Messungen (Verbrauch, Umwandlungen usw.)
- Erstellung personalisierter Anzeigen (Kombinationen von Variablen, Fehlergruppen usw.)
- Definition neuer Automatisierungstypen für die Steuerung oder Überwachung (Umstellung usw.)

Eine Formel ist mit einer Variablen verknüpft.
Nach ihrer Ausführung weist die Formel den erhaltenen Wert der zugeordneten Variablen zu.
Ihr Wert wird bei jeder Berechnung anhand der Formel gemäß einer konfigurierbaren Zykluszeit aktualisiert (Mindestzykluszeit = 100 ms)

2.2 Syntax und Regeln

Die Formeln verwenden eine natürliche und intuitive Syntax, die Referenzen mit Messwerten (Variablen, historische Daten usw.), Ausdrücken aus Buchstaben (Konstanten, Operatoren usw.) und Funktionsaufrufen (min, max, delta usw.) kombiniert.
Der Benutzer muss sich nicht um Datenformate kümmern: Der Interpreter konvertiert Datentypen automatisch. Dadurch wird die Kombination von logischen und mathematischen Verknüpfungen möglich: *(B > 3)** C, *(B>3)* ist ein logischer Ausdruck (wahr (1) oder falsch (0)).

Die Formeln machen nur Sinn, wenn am Ende ihrer Ausführung der erhaltene Wert einer Variablen (oder einer Datenbankzelle) zugeordnet wird. Bitte denken Sie daran, dass **eine Formel keine Skriptsprache ist**: Schleifen (for, while) oder Sprungbefehle (goto, if.. then) sind nicht vorgesehen. Es gibt jedoch einen Konditionaloperator, der die Zuordnung von Werten gemäß logischen Bedingungen ermöglicht.

Die Formeln berücksichtigen die Schreibweise (Groß- oder Kleinbuchstaben) der Operanden und Funktionen. Die Formellänge ist auf 200 Zeichen begrenzt.
Eine Formel kann bis zu 50 Elemente enthalten (Operanden, Operatoren…).

Eine Formel kann in verschiedene „Unterformeln“ unterteilt sein, die durch ein **Semikolon voneinander getrennt werden (das Komma ist für die Abtrennung eines Funktionsparameters reserviert)**.
Alle „Unterformeln“ werden innerhalb eines Bearbeitungszyklus nacheinander in der Reihenfolge des Schreibens ausgeführt.

2.2.1 Operanden

2.2.1.1 Numerische Konstanten (Ganzzahlen, Dezimalzahlen):

Die Zahlen können positiv oder negativ sein, es können Ganz- oder Dezimalzahlen bzw. Exponenten sein: 123, -45.1, 12.5E3, usw.
Um einen Hexadezimalwert einzugeben, muss dem Wert das Symbol $ (oder der Buchstabe H) vorangestellt werden, zum Beispiel $10 (oder H10).
Bei „Dezimal“-Werten wird als Dezimaltrennzeichen der Punkt verwendet.

Die Art des Werts (Ganzzahl, Wort, IEEE) wird durch die Analyse der Formel automatisch ermittelt.

2.2.1.2 Zeichenketten:

Eine Zeichenkette muss von einfachen Anführungszeichen (') eingefasst sein, zum Beispiel '*meine Kette*'.
Zeichenketten werden zum Verweis auf bestimmte Funktionsparameter oder Namen von Tabellen- oder Datenbankspalten verwendet.

2.2.1.3 Variablen:

Durch Eingabe des Namens einer **Variablen** (ohne Anführungszeichen) wird der Operand zum Ausführungszeitpunkt als **Wert der Variablen** ausgelegt. Die Art des Werts entspricht der Variablen.

Warnung: **Der Name** einer mit einer Formel verwendeten Variablen **darf keine Leerzeichen enthalten.**

2.2.2 Operatoren

2.2.2.1 Mathematisch:

+	Addition
-	Subtraktion
*****	Multiplikation
/	Division (eine Teilung durch Null verursacht einen Ausführungsfehler)
%	Modulo: Rest der Gesamtdivision (ganze Zahlen)
******	Leistung: *A***B* = *A* **Exponent** *B* = A^B

2.2.2.2 Logisch:

&&	Logisch UND	*A* **&&** *B* = wahr, wenn *A* = *wahr* **UND** *B* = wahr, ansonsten *falsch*
## **||**	Logisch ODER	*A* **||** *B* = *wahr*, wenn *A* = *wahr* **ODER** *B* = wahr, ansonsten *falsch*
^^	Logisch EX-KLUSIV ODER	*A* **^^** *B* = *wahr,* wenn (*A* = *wahr* **UND** *B* = *falsch*) **ODER** (*A* = *falsch* **UND** *B* = *wahr*)
!	Logisch NICHT	**!** *A* = *wahr,* wenn *A* = *falsch*

Anmerkung: Das Symbol „|“ wird bei gleichzeitigem Drücken der Tasten [AltGr] und [6] auf der Tastatur angezeigt.

2.2.2.3 Binär:

&	Bitweise UND	*$AA* **&** *$55* = *$00*
I **#**	Bitweise ODER	*$AA* **|** *$55* = *$FF*
^	Bitweise EXKLUSIV ODER	*$AA* **^** *$A5* = *$0F*
~	Bitweise NICHT (ergänzend zu 1)	**~***A,* negiert alle *A*-Bits: 1 wird zu 0 und umgekehrt
<<	Verschieben nach links	*A* **<<** *B,* verschiebt die *A*-Bits um *B-Bits* nach links
>>	Verschieben nach rechts	*A* **>>** *B,* verschiebt die *A*-Bits um *B-Bits* nach rechts

2.2.2.4 Vergleiche:

= **==**	Gleich
!= **<>**	Verschieden
>	Größer als
<	Kleiner als
>=	Größer als oder gleich
<=	Kleiner als oder gleich

2.2.2.5 Zuordnungsoperator

Der Operator „:=“ dient der Zuordnung eines Werts zu einer Variablen.
Beispiel: „*mavar:= 2*".
Der Benutzer muss sich nicht um das Ziel-Datenformat kümmern, da die Umwandlung automatisch erfolgt.
Bei der Ein-/Ausgabe einer Peripherievariablen wird durch die Zuordnung beim Ausführen der Formel ein Schreibbefehl an die Peripherievariable gesendet.

2.2.2.6 Konditionaloperator

Der Operator (expr)? A: B übergibt A, wenn expr wahr ist, ansonsten B.
expr muss ein boolescher Ausdruck sein.
Beispiel:
A:= (B>C)? D: E; weist D zu A, wenn B größer als C ist, ansonsten wird E A zugewiesen.
Es können natürlich auch Klammern verwendet werden, um komplexere Ausdrücke zu erstellen.
Das gleiche Prinzip gilt für die Operationen:
A + ((B>C)? D: E) ist A + D wenn B > C, ansonsten ist es A + E.
Der Konditionaloperator kann auch für die Zuordnung verwendet werden:
((B > C)? D: E):= A;
Wenn B > C, wird D der Wert A zugeordnet, ansonsten wird E der Wert A zugeordnet.

2.2.2.7 Reihenfolge der Bedeutung von Operatoren und Verwendung von Klammern

Prioritätenfolge bei der Ausführung der Operationen eines Ausdrucks:

Funktion	Fcn ( )	P0 = Höchste Priorität
Positivzeichen	+ (Var)	P1
Negativzeichen	- (Var)	
Logisch NICHT	!	
Ergänzend zu 1 *	~	
Leistung	**	P2
Multiplikation	*	P3
Division	/	
Rest *	%	
Addition	+	P4
Subtraktion	-	
Umschalten links *	<<	
Umschalten rechts *	>>	
Bitweise UND *	&	P5
Bitweise ODER *	I/#	
Bitweise EXKLUSIV	^	
Logisch UND	&&	P6
Logisch ODER	II/##	
Logisch EXKLUSIV	^^	
Test gleich wie	=	
Test kleiner oder gleich	<=	
Test größer oder gleich	>=	
Test kleiner als	<	
Test größer als	>	
Test verschieden von	!=/<>	
Zuordnung	:=	P7 = Min. Priorität

„*“: Nur Operation mit Ganzzahl (Verarbeitung mit Bits)

2.3 Funktionen

Das Funktionsargument kann jegliche Art von Operand sein, aber auch mathematische oder logische Ausdrücke oder Funktionsergebnisse. Somit lassen sich mehrere Funktions- und Operationsaufrufe miteinander verknüpfen.

„, …)“: die drei kleinen Punkte stehen für Funktionen mit einer wechselnden Zahl an Argumenten.
„[x]“: die rechteckigen Klammern beziehen sich auf ein (oder mehrere) Operationsargument(e).

3.1 Mathematische Funktionen

Außer bei anderslautenden Angaben ist das von diesen Funktionen zurückgegebene Werte-Datenformat IEEE (floating simple precision).

2.3.1.1 abs(x)

Gibt den absoluten Wert seines Arguments x zurück.
Der von der Funktion zurückgegebene Datentyp ist der gleiche wie der von x.
Beispiel: Abs(VAR1*3)

2.3.1.2 sqrt(x)

Gibt die Quadratwurzel von x zurück.
Wenn x kleiner Null ist, verursacht die Operation einen Ausführungsfehler.

2.3.1.3 log(x)

Gibt den napierischen Logarithmus von x zurück.
Wenn x kleiner oder gleich Null ist, verursacht die Operation einen Ausführungsfehler.

2.3.1.4 log10(x)

Gibt den Logarithmus als Basis 10 von x zurück.
Wenn x kleiner oder gleich Null ist, verursacht die Operation einen Ausführungsfehler.

2.3.1.5 exp(x)

Gibt den Exponentialwert von x zurück. exp(log(x)) = x.

2.3.1.6 pow(x, y)

Gibt x zu der Leistung von y zurück. pow(x, y) = x ** y = xy.

2.3.1.7 intg(x)

Gibt die ganze Zahl von x zurück. Zum Beispiel: intg(12.46) = 12.
Der zurückgegebene Wert ist eine Ganzzahl (INT32).

2.3.1.8 rand(x)

Gibt eine zufällige Ganzzahl zurück, die größer als oder gleich 0 und kleiner x ist.
Der zurückgegebene Wert ist ein Doppelwort (DWORD).

2.3.2 Statistische Funktionen

2.3.2.1 mini(A, B, …)

Gibt das Argument mit dem geringsten Wert zurück.
Diese Funktion muss **mindestens zwei** Argumente umfassen.

Es ist zu beachten, dass diese Funktion nicht nur den Wert des Arguments, sondern auch seine Referenz zurückgibt. Das bedeutet, dass es mit einer Zuordnungsoperation kombiniert werden kann:
mini(*VAR1***,** *VAR2***,** *VAR3)***:=** *4*; mit *VAR1*, *VAR2*, *VAR3* der Variablenreferenzen, wenn *VAR3* den Mindestwert hat, wird diese Variable dem Wert **4** zugeordnet.

2.3.2.2 maxi(A, B, …)

Gibt das Argument mit dem größten Wert zurück.
Diese Funktion muss **mindestens zwei** Argumente umfassen.

Wie bei **mini** gibt diese Funktion die Argumentreferenz zurück.

2.3.2.3 pmini(A, B, …)

Gibt die Position (von 0) des Arguments mit dem geringsten Wert zurück.
Diese Funktion muss **mindestens zwei** Argumente umfassen.

2.3.2.4 pmaxi(A, B, …)

Gibt die Position (von 0) des Arguments mit dem größten Wert zurück.
Diese Funktion muss **mindestens zwei** Argumente umfassen.

2.3.2.5 sum(A, B, …)

Gibt den Wert der Summe der Argumente zurück.
Diese Funktion muss **mindestens zwei** Argumente umfassen.

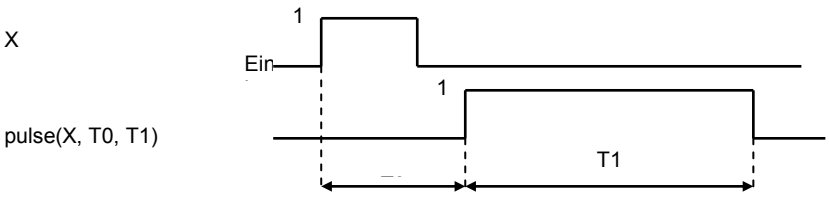
2.3.2.6 avg(A, B, …)

Gibt den Wert des Argumentdurchschnitts zurück.
Diese Funktion muss **mindestens zwei** Argumente umfassen.

2.3.3 Logische Funktionen

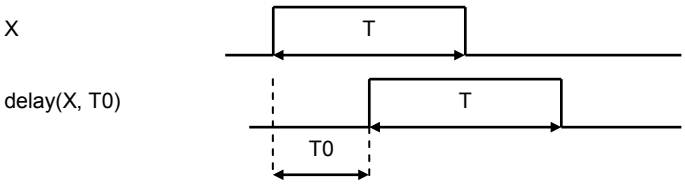
2.3.3.1 pulse(X, T0, T1)

X ist ein boolescher Wert. T0 und T1 sind Werte in Sekunden.
An der Vorderflanke von X (Sprung von 0 auf 1) liefert die Funktion nach T0 Sekunden für die Dauer von T1 Sekunden TRUE (1).
Die restliche Zeit liefert sie FALSE (0).



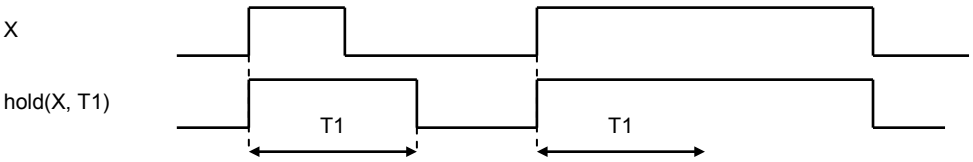
2.3.3.2 delay(X, T0)

X ist ein boolescher Wert. T0 ist ein Wert in Sekunden.
Die Funktion kopiert den Wert von X und wird um T0 Sekunden verschoben.



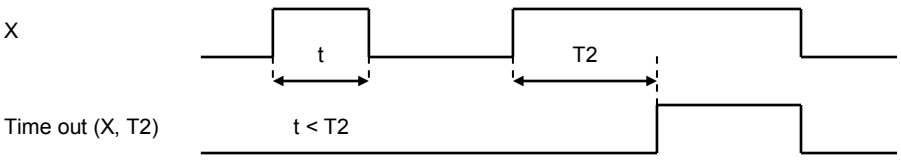
2.3.3.3 hold(X, T1)

X ist ein boolescher Wert. T1 ist ein Wert in Sekunden.
Die Funktion liefert für mindestens T1 Sekunden an einer Vorderflanke von X (Sprung von 0 auf 1) TRUE. Bleibt X für mehr als T1 Sekunden 1 (TRUE), liefert die Haltefunktion während dieser Zeit TRUE.



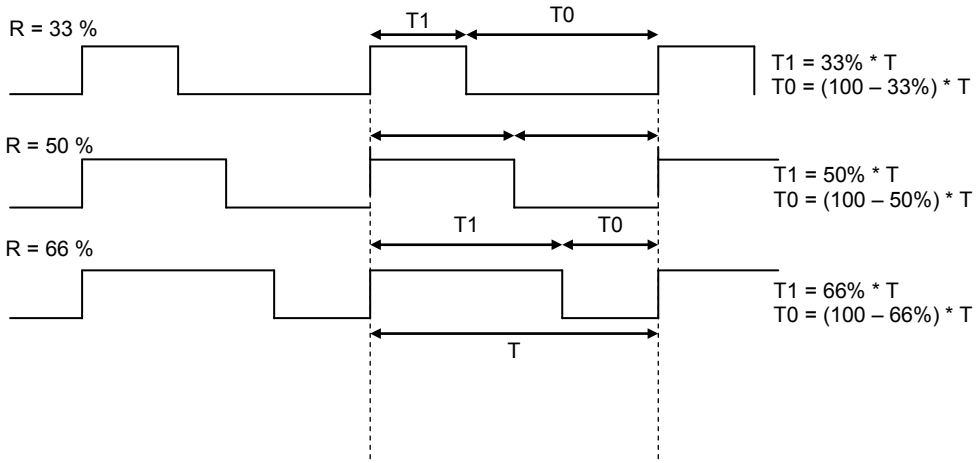
2.3.3.4 tempo(X, T2)

X ist ein boolescher Wert. T2 ist ein Wert in Sekunden.
Die Funktion liefert TRUE, wenn X für mindestens T2 Sekunden auf 1 wechselt. Nach dieser Zeit liefert die Funktion so lange TRUE, wie X auf TRUE bleibt.



2.3.3.5 pwm(T, R)

Die Funktion gibt einen booleschen Wert zum Rhythmus eines PWM-Signals der Dauer T und dem zyklischen Verhältnis R als Prozentzahl zurück. R muss zwischen 0 und 100 % sein.



2.3.3.6 iswithin(X, min, max)

gibt einen booleschen Wert zurück, der wahr ist, wenn (X >= min) **UND** (x <= max). Ansonsten ist er falsch.

2.3.4 Zeit/Datum und Zeitfunktionen:

Das Format TIME_T ist ein Datenformat, das der Darstellung von Datum/Uhrzeit als Anzahl von Sekunden bezogen auf die Referenz (1. Januar 1970 um 00:00:00) dient.
Den Wochentagen sind folgende Werte zugewiesen:
1 = Sonntag, 2 = Montag, 4 = Dienstag, 8 = Mittwoch,
16 = Donnerstag, 32 = Freitag, 64 = Samstag.

2.3.4.1 dt([[T], X]) oder now([[T], X])

Ohne ein Argument gibt die Funktion das aktuelle Datum und die Uhrzeit als TIME_T zurück.

Mit T als TIME_T und X dient T als Referenz für das Argument X.
Nur mit X dienen die aktuellen Datums- und Uhrzeitfunktionen als Referenzen für das Argument X.
Wenn X = 1 oder „Y“ wird das Jahr in 4 Ziffern angegeben.
Wenn X = 2 oder „M“ wird der Monat angegeben (von 1 bis 12).
Wenn X = 3 oder „D“ wird der Tag des Monats angegeben (von 1 bis 31).
Wenn X = 4 oder „H“ wird die Uhrzeit gemäß der 24-Stunden-Uhr angegeben (von 0 bis 23).
Wenn X = 5 oder „m“ wird die Minute angegeben (von 0 bis 59).
Wenn X = 6 oder „s“ wird die Sekunde angegeben (von 0 bis 59).
Wenn X = 7 oder „d“ wird der Tag der Woche angegeben (1, 2, 4, 8,16, 32, 64).
Wenn X = 8 oder „p“, wird 0 für die Zeit vor Mittag (AM), und 1 für den Nachmittag (PM) zurückgegeben.
Wenn X = 9 oder „h“ wird die Uhrzeit gemäß der 12-Stunden-Uhr angegeben.
Wenn X = 10 oder „y“ wird das Jahr in 2 Ziffern angegeben.

2.3.4.2 time(X [, F])

Nur mit X, wenn X eine Zeichenkette ist, werden Datum und Uhrzeit von X gemäß Standarduhrzeit und Zeitformat von der Funktion in das Format TIME_T umgewandelt. Wenn X numerisch ist, wandelt die Funktion diese Zahl in TIME_T um.
Mit X und F muss X eine Zeichenkette sein. F ist auch eine Zeichenkette, die das Leseformat für Datum und Uhrzeit definiert. Die Funktion gibt daher Datum X als TIME_T gemäß Format F zurück.

Für F gilt das Format „**d/m/y H:M:S**“ mit:
d = Tag des Monats (von 1 bis 31)
m = Monat des Jahres (von 1 bis 12)
y = das Jahr in 2 Ziffern
Y = das Jahr in 4 Ziffern
H = die Uhrzeit gemäß der 24-Stunden-Uhr
h = die Uhrzeit gemäß der 12-Stunden-Uhr
M = die Minute (von 0 bis 59)
S = die Sekunde (von 0 bis 59)

2.3.4.3 isweek([X])

Wenn X (optional) nicht vorhanden ist, gibt die Funktion den booleschen Wert wahr zurück, wenn das aktuelle Datum ein Werktag ist (Montag bis Freitag).
Wenn X der Typ TIME_T ist, wird die Funktion wahr zurückgegeben, wenn das Datum X ein Werktag ist. **Ansonsten** bezieht sich X auf eine Zahl für den Tag der Woche und die Funktion wird wahr zurückgegeben, wenn Tag X ein Werktag ist.

2.3.4.4 deltats()

Gibt die Zeit in SEKUNDEN zwischen zwei Ausführungen der Formel zurück.
Diese Funktion ist sehr leistungsfähig, da sie die Erstellung von Integrationen, Differentialkoeffizienten usw. ermöglicht.
Sie basiert auf der Echtzeituhr des Systems.

2.3.4.5 deltatms()

Gibt die Zeit in MILLISEKUNDEN zwischen zwei Ausführungen der Formel zurück.
Diese Funktion ist sehr leistungsfähig, da sie die Erstellung von Integrationen, Differentialkoeffizienten usw. ermöglicht.
Sie basiert auf der Quarzuhr der CPU. Das heißt, dass sie langfristig relativ ungenau ist.

2.3.5 Sonstige Funktionen:

2.3.5.1 bounds(X, min, max)

Gibt den Wert *X* begrenzt durch *min* und *max* zurück.
Wenn *X* kleiner ist als *min*, wird *min* zurückgegeben; wenn *X* größer ist als *max*, wird *max* zurückgegeben, ansonsten wird *X* zurückgegeben.

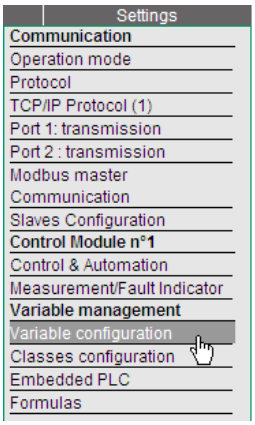
2.3.5.2 delta(X) oder deltav(X)

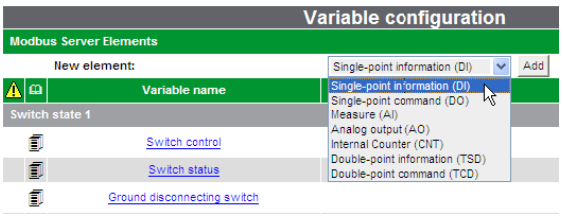
Gibt die Differenz zwischen dem Wert von X bei der vorherigen Ausführung der Formel und des aktuellen Werts von X an. Der Zeitraum für die Berechnung der Differenz bei X ist daher die Ausführungszeit der Formel. Siehe ebenso deltats() und deltatms().

2.3.5.3 switch(X, A0, A1 [, A2, …]) oder cond(X, A0, A1 [, A2, …])

X muss ein positiver ganzer Wert sein.
Dieser Funktion müssen **mindestens drei** Argumente übergeben werden.

Wenn X = 0, gibt die Funktion das Argument A0 zurück, wenn X = 1, wird A1 zurückgegeben, wenn X = 2, wird A2 zurückgegeben, usw.
Wenn X < 0, gibt die Funktion das Argument A0 zurück, wenn X größer als die Zahl der Argumente ist, wird das aktuellste zurückgegeben.





Seite „*Variable configuration*“

2.4 Einstellungen

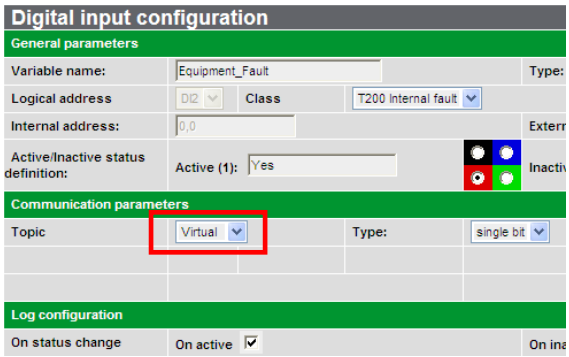
2.4.1 Variable erstellen

Der Benutzer muss Variablen deklarieren, die speziell den „Formeln“ zugeordnet sind. Sie werden nur von der „Formeln“-Funktion aktualisiert, werden aber wie jede andere Variable in Bezug auf Protokoll und Statusanzeige verwaltet.

Die Erstellung wird auf der Seite „*Variable Configuration*“ des Webservers ausgeführt, auf die gleiche Weise wie für die anderen Variablentypen.

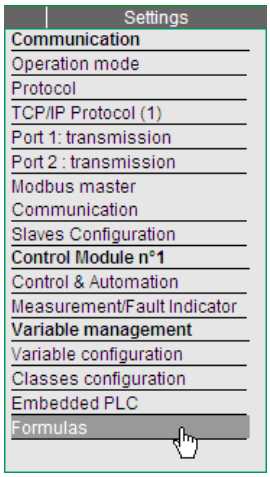
Für die Funktionen können folgende Variablentypen erstellt werden:

Variablen-typ	**Verwendung**	**Beschreibung**
DI	Einzel-Fernsignal	Einzel-Bit-Anzeige
DO	Einzel-Fernbefehl	Einzel-Bit-Befehl
AI	Analogmessungen	Von dem T200/F200C durchgeführte Analogmessungen (Strom, Spannung usw.)
AO	Analogbefehl	Analogschreiben
CNT	Zähler	Änderung des Zustandszählers bei Einzel-Bit
TCD	Doppel-Fernbefehl	Doppelbefehl an Leistungs-/ Trennschalter
TSD	Doppel-Fernsignal	Anzeige der Leistungs-/ Trennschalter-Position



Der Name der den Berechnungsformeln zugeordneten Variablen darf keine Leerstelle enthalten.

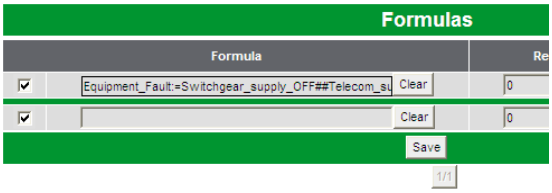
Eine der Formel zugeordnete Variable muss mit einem „*Virtuellen*“ Topic konfiguriert sein.



2.4.2 Formeleingabe

Sobald die Variable erstellt wurde, muss die zugeordnete Formel in die Seite „Formulas“ des Webservers eingegeben werden.

Diese Seite kann bis zu 100 Formeln enthalten. Es können 20 Formeln gleichzeitig angezeigt werden.

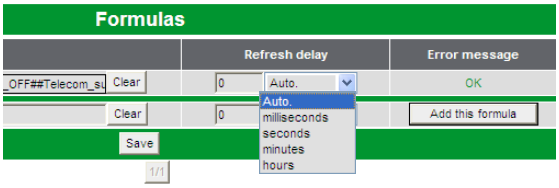


Beispiel einer definierten Formel

Regeln für die Formeleingabe:

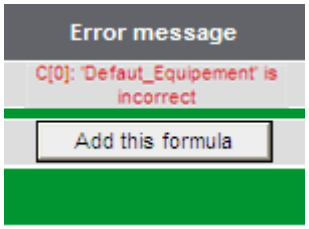
- Jede Formel muss folgendermaßen beginnen: „name_of_the_variable:=“
- Im Namen der Variablen darf kein Leerzeichen sein.
- Der Inhalt der Formeln mit den Operanden und Operatoren wird nach dem Zeichen „…:=“ definiert.
- Klein- oder Großschreibung muss bei den Namen der Variablen und Operanden beachtet werden.

Jede Formel kann über ein Kontrollkästchen aktiviert/deaktiviert werden.



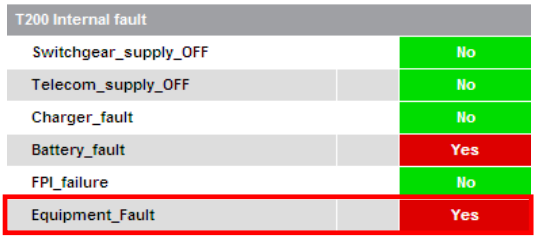
Es muss ein Zeitraum für die Aktualisierung der Formel angegeben werden. Ein als „Auto“ definierter Zeitraum entspricht dem kürzest möglichen Zeitraum (100 ms).

Auffrischungsverzögerung einer Formel



Wenn die Syntax der Formel falsch ist, wird eine Fehlermeldung angezeigt.

Beispiel einer Fehlermeldung bei falscher Eingabe in eine Formel



Das Ergebnis der Formel kann auf der Seite „*Monitoring*“ angezeigt werden.

Es entspricht dem Zustand der Variablen, die der Formel zugeordnet ist, die zu ihrem Ergebnis gehört.

Formelergebnis in Seite „Monitoring“

Persönliche Notizen

Persönliche Notizen

Schneider Electric Industries SAS

Schneider Electric Telecontrol
839 Chemin des Batterses
Z.I. Ouest
01700 St Maurice de Beynost
Tel: +33 (0)4 78 55 13 13
Fax: +33 (0)4 78 55 50 00

http://www.schneider-electric.com
E-Mail: telecontrol@schneider-electric.com

Aufgrund der ständigen Normen- und Geräteentwicklungen sind die hier aufgeführten technischen Eigenschaften und Darstellungen erst nach firmenseitiger Bestätigung verbindlich.

Dieses Dokument wurde auf umweltfreundlichem Papier gedruckt.

NT00320-DE-02 10/2014

Veröffentlichung, Produktion und Druck: Schneider Electric Telecontrol
Made in France - Europa